



**Fermi National Accelerator Laboratory**

**FERMILAB-Conf-91/241**

# **Tracking and Vertex Finding with Drift Chambers and Neural Networks**

**C. Lindsey**

*Fermi National Accelerator Laboratory  
P.O. Box 500, Batavia, Illinois 60510*

**September 1991**

\* Presented at the *Workshop on Neural Networks: From Biology to High Energy Physics*, Isola d' Elba, Italy,  
June 5-14, 1991.



Operated by Universities Research Association Inc. under contract with the United States Department of Energy

# Tracking and Vertex Finding with Drift Chambers and Neural Networks \*

Clark S. Lindsey  
*Fermi National Accelerator Laboratory* †  
P.O. Box 500  
Batavia, Illinois 60510  
USA

## ABSTRACT

Finding tracks, track vertices and event vertices with neural networks from drift chamber signals is discussed. Simulated feed-forward neural networks have been trained with back-propagation to give track parameters using Monte Carlo simulated tracks in one case and actual experimental data in another. Effects on network performance of limited weight resolution, noise and drift chamber resolution are given. Possible implementations in hardware are discussed.

## 1. Introduction

Determining track parameters from drift chamber signals with neural networks has been discussed previously. Both Hopfield<sup>1,2</sup> and feed-forward type networks<sup>3</sup> have been used. The work here is aimed towards applications of neural networks in fast trigger systems. Feed-forward networks are therefore emphasized since they are now becoming available in VLSI hardware. In such trigger schemes, the drift chamber signals would be fed directly into neural nets which would then pass tracking information to higher level trigger processors (perhaps also composed of neural nets) which in turn would decide if an event is to be recorded on tape.

High energy experiments make extensive use of drift chambers to track charged particles. The particles ionize molecules in a gas and the resulting free electrons drift towards a *sense* wire at a high positive voltage. When the electrons approach close to the wire, the high electric field there causes an avalanche of ionization which produces a measurable signal at the ends of the wires. Field wires and/or planes at negative or ground voltages separate the sense wires and help shape the electrostatic fields. Measuring the time between when the track crossed the chamber and when the signal appears gives the electron drift time. By knowing the velocity of the electrons in the gas, one can calculate the drift distance. In turn, knowing the coordinate

---

\*Invited talk presented at the *Workshop on Neural Networks: From Biology to High Energy Physics*, Isola d'Elba, Italy, June 5-14, 1991

†Fermilab is operated by the Universities Research Association under contract with the Department of Energy

of the wire gives a space point. There are typically many layers of sense wires to provide several space points along the tracks.

We discuss two applications of neural nets to drift chamber tracking. In the first case we simulate particles traversing a three layer drift chamber that is patterned after the Tevatron D0 experiment muon chambers. The neural net uses the resulting drift times to determine the slopes and intercepts of the tracks. In the second case, we used data from an experiment at the Tevatron  $p\bar{p}$  collider to train a net to determine the vertex (i.e. intercept of a track with the beam line) of a track traversing a section of a chamber adjacent to the beam line. Combining the vertices found for all sections then provides the event vertex of the  $p\bar{p}$  collision.

## 2. Muon Chamber Tracking

Many detector systems for colliding beam experiments have sets of drift chambers for detecting high energy muons emitted from the interactions. The muon system usually consists of several chambers separated by large slabs of iron which filter out most particles except for muons. Because of the large area they must cover, the chambers usually only have 3 or 4 layers of sense wires to minimize cost.

Here we investigate the possible use of neural networks to find track slope and intercept in a typical muon chamber design. This could be useful for fast identification of high momentum muons pointing back towards the interaction vertex. Events with such tracks are often of great interest.

Figure 1 shows three examples of simulated tracks in a section of a drift chamber fashioned after the design being used in the D0 experiment at the Tevatron  $p\bar{p}$  collider.<sup>4</sup> Here only 6 sense wires are shown and the field and ground planes are omitted. The wires are spaced apart about 10cm horizontally and about 2.5cm vertically. The cells in a layer are combined into pairs such that two sense wires are connected electrically at one end. The signal from one sense wire goes to the stop of a time to voltage converter (TVC) which gets its start from the beam crossing timing. The TVC output voltage is therefore proportional to the drift time. One sense wire also sends a stop signal to a second time to voltage converter which gets its start from the other sense wire. This dTVC output voltage is proportional to the position along the sense wires where the ionization occurs. Although the dTVC signal indicates which cell of a pair the track passed, there is still an ambiguity in determining which side of the sense wire the track crossed. The cells in the layers are staggered with respect to one another to help resolve this *left-right* ambiguity.

A neural network was designed to find the slope and intercept of tracks passing through a set of 6 cells (or 3 pairs). Although the dTVC gives information in the dimension along the sense wire, the tracking here is only done in the plane normal to the wires. The dTVC simply indicates through which cell of the pair the track passed.

Figure 2 shows the architecture of the net. The three TVC voltages and the three dTVC voltages are inputs to the net. There are 64 hidden and 64 output units. The output units are divided into two sets of 32 units, with one set providing

the intercept of the track in the plane of the middle layer of wires and the other set giving the slope of the track. Each output neuron corresponds to a *bin* in intercept or slope. The net will be trained to excite only those output neurons which correspond to the slope and intercept of the track. The target patterns are given as Gaussian activations across 3 or 4 neurons so that the calculated average of the slope and intercepts can be more precisely calculated than just the width of a single bin.

A standard back-propagation algorithm was used to train the net.<sup>6</sup> The units had sigmoid transfer functions of the type

$$f_j(x_j) = \frac{1.0}{1.0 + \exp(-x_j)}$$

where  $x_j$  is

$$x_j = \sum_k w_{jk} V_k + w_{jb} V_{bias}.$$

Here  $V_k$  is the output voltage of unit  $k$  in the preceding layer and  $w_{jk}$  is the weight for the connection between units  $j$  and  $k$ . A bias voltage and weighting is also provided.

The net was trained for several hundred passes through a set of 10000 patterns. Both the TVC and dTVC ranged from 0.0 to 3.5v. The training input values were derived from the exact drift distances. Results here are given, unless otherwise noted, for a test set with the drift distances smeared with a Gaussian distribution of sigma equal to  $400\mu m$ . This roughly simulates the drift resolution of the actual chambers.

Figure 1 shows the generated tracks and the tracks found by the net. Also shown are the excitations of the net output units. The generated track parameters are compared to the net outputs. The net value is taken as the average over the range of  $\pm 2$  units around the bin with the largest excitation. For these cases the net came quite close to the generated track. Notice in the output excitations that there are sometimes units excited due to ambiguities in the tracking. This problem occurs because with only 3 layers, the staggering of the cells is not always sufficient to resolve the ambiguities, especially for tracks at wide angles with respect to vertical.

Figure 3 shows distributions of the differences in the parameters between the target values and the network for the test sets. The sigmas of the Gaussian fits are  $490\mu m$  and  $0.014\text{rad}$  ( $= 0.8^\circ$ ) for the intercept and slope respectively. Because of the tracking ambiguities the distributions are not exact Gaussians. Over 95% of the entries are within 3 sigma of the peaks.

Although the net was trained on patterns without drift smearing, the above results show that the net still performs well on patterns with smearing. Figure 4 shows the dependence of the tracking accuracy of the net on the drift distance smearing. The net resolution sigmas are weakly dependent on the drift smearing at first but then begin to increase linearly with increase in drift smearing.

As will be discussed in the last section, implementation of the net in hardware could require a decrease in the precision of the weight values (largest weight values were around 10.0). To investigate the effect of degraded weight precision, we truncated the weight values and then determined the net resolution for the test set.

The table shows the effect on the resolution of truncating the fractional part of the weight from the full floating point value down to integer values. Only for the integer case do the results become substantially worse. However, if "chip in the loop" training can be done, even integer weights might be sufficient.

Truncation of Weights vs Tracking Resolution		
Weight	Intercept Resolution	Slope Resolution
Full precision	490 $\mu m$	0.80°
2 decimal places	500 $\mu m$	0.80°
1 decimal place	550 $\mu m$	0.92°
0 decimal places	2900 $\mu m$	3.4°

### 3. Vertex Finding

In proton-antiproton collision experiments, the intercept of a track at the beam line is taken as the track *vertex* or origin. The primary tracks in an event originate from the point where the initial interaction occurred. This point is referred to as the *event vertex*. Tracks in the detector may also have come from decays of primary particles and from interactions of the primaries with intervening materials. The origins of these tracks are called secondary vertices.

In a manner similar to the track intercept finding method discussed in section 2, a net can be trained to find the intercept of the track at the beam line instead of in a plane of the chamber. The position along the beam line where the majority of these track vertices originated in a given event is taken as the best estimate for the event vertex position. Fast determination of the event vertex could be useful, for example, for SSC experiments where there will be two or three pp interactions in every beam crossing.

A small 3-layer drift chamber (called the z-chamber) was placed beside the beam pipe in experiment E735 at the Tevatron proton-antiproton collider to find the track and event vertex positions along the beam line. Using actual data taken with the z-chamber, a net was trained to find track vertices in subsections of the chamber. Adding the outputs of all the subsection nets then provided the position of the event vertex. The results of this study are discussed in detail in reference 5 so only a brief review will be given here.

Figure 5 shows a schematic drawing of an event and the resulting hits in the z-chamber. Since the chamber had 288 wires, it was broken up into smaller subsections of 18 wires each (3 layers, 6 wires per layer) so that the net could be of manageable size. The drift times (normalized to 1.0) were fed into a net of 18 inputs, 128 hidden units and 64 output units. Figure 6 shows the architecture of the subsection net. In a similar manner to the muon chamber net, the output units corresponded to bins over a range of  $\pm 30$ cm from the center of each subsection. The target values (as calculated by a conventional tracking algorithm) were given as Gaussians over 3 or 4 bins with sigma of 0.5cm.

Figure 7 shows examples of tracks in the subsection nets and the resulting net outputs. Often the net finds the track vertex even when there are extraneous noise or background hits or more than one track. Note that some output units are activated at positions corresponding to "alternative" choices due to the left-right ambiguities. The tracking resolution in the z-chamber was  $500\mu\text{m}$  and the chamber vertex resolution was 0.5cm. Figure 8 shows distributions of the differences between the fit vertex and the net vertex for cases of one track, one track plus background hits and two tracks in the subsection. The resolution degrades with increases in number of hits because of the problem with ambiguities.

The overlapping subsections were spaced every two cells. In a hardware implementation each subsection would have its own net and the processing would be done in parallel. Here the drift times of each subsection were fed sequentially into the net simulation and the net output values added to the cumulative distribution. After all the subsections were processed, the peak in the resulting distribution was taken as the event vertex position. Figure 9 shows some examples of whole events and the cumulative net distributions. Note that the cumulative distribution again gives alternative peaks both because of ambiguities and to the fact that not all tracks point to the same vertex. Secondary tracks, such as those from interactions of the primaries with beam pipe material, will usually not point back to the event vertex. As much as 40% of the tracks in an event can be from secondary interactions. The z-chamber only covered a small portion of the total solid angle so it saw only a small sample of the tracks. This causes considerable fluctuations in the ratio of primary tracks to secondary tracks in the z-chamber. Figure 9(d) shows an example where only one of 5 or 6 tracks in the chamber point to the vertex chosen by the fitting algorithm as the best choice for the event vertex while the neural net method chose a different vertex. The method here should improve if used in chambers that are bigger and sample a larger number of tracks.

The net event vertices are compared to that obtained from the conventional off-line track fitting methods and to the "time-of-flight" vertex. The latter system uses the hit times of particles traversing scintillators near the beam pipe to find the event vertex. The TOF system is not as accurate as tracking but its signals are available for fast trigger applications. Figure 10 shows distributions of the differences between the tracking vertex and the net vertex and the TOF vertex. The net gives a sigma of about 1.4 cm and is within 10 cm of the fit vertex 89% of the time. This compares favorably to the TOF method which gave 3.8 cm resolution and had vertices within 10 cm of the fit vertex 83% of the time.

#### 4. Discussion

We have shown that a neural network can be trained to give track parameters from drift chamber signal inputs. Both simulated and experiment data have been fed to neural net simulations and they perform well despite finite drift resolution, noise, limited weight precision, etc.

The "histogram" manner used here for encoding the tracking parameters in the

output units of the network provides (1) a way for the net to give possible alternative answers in those cases where there are ambiguities, and (2) allows a straight forward way to combine outputs from separate sections of the detectors (somewhat similar to the histogramming method of pattern recognition.) Using only a single neuron whose value was proportional to the slope, for example, would obviously reduce the size of the net required but it would not have these two advantages.

Work is now underway to implement in hardware the kinds of networks discussed here. We are currently experimenting with the Intel ETANN chip.<sup>7</sup> This analog chip has 64 inputs (optionally 128 inputs can be clocked through sequentially in two sets of 64 inputs) and 16 bias units totally interconnected to 64 neurons. The chip has two sets of 80x64 weights of about 6 to 8 bits precision. Signals propagate through the first set of weighted connections to the neurons whose outputs are either available directly or can be clocked back through the second set of weights to give 3 layer performance by reuse of the neurons. One such chip could handle the muon tracking net and 3 chips (one in 128 input mode) could do the z-chamber track vertex finding. If the current tests are successful, we hope to begin developing a neural network tracking system in the near future.

### Acknowledgements

This work was done in collaboration with Bruce Denby of Fermilab and with the D0 muon chamber group, including Ken Johns (Arizona), Herman Haggerty (Fermilab), and Mike Fortner (Northern Illinois State). We thank the Wisconsin z-chamber group including Theo Alexopoulos and Al Erwin. This research was supported by Fermi National Accelerator Lab.

### References

1. B. Denby, *Computer Phys. Commun.* **49** (1988) 429.
2. C. Peterson, *Nucl. Inst. & Meth.* **A279** (1989) 537.
3. B. Denby, E. Lessner and C. S. Lindsey *Proc. 1990 Conf. on Computing in High Energy Physics*, Sante Fe, NM. 1990, AIP Conf Proc. **209**, 211.
4. C. Brown et al., *Nucl. Inst. & Meth.* **A270** (1989) 331.
5. C.S. Lindsey and B. Denby, *Nucl. Inst. & Meth.* **A302** (1991) 217.
6. D. Rumelhart et al., *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, vol. I, chap. 8, (MIT Press, Cambridge, Ma, 1986).
7. M. Holler et al., *Proc. Int. Joint Conf. on Neural Networks*, Washington, D.C., 1989, vol. II, IEEE Catalog 89CH2756, p. 191.

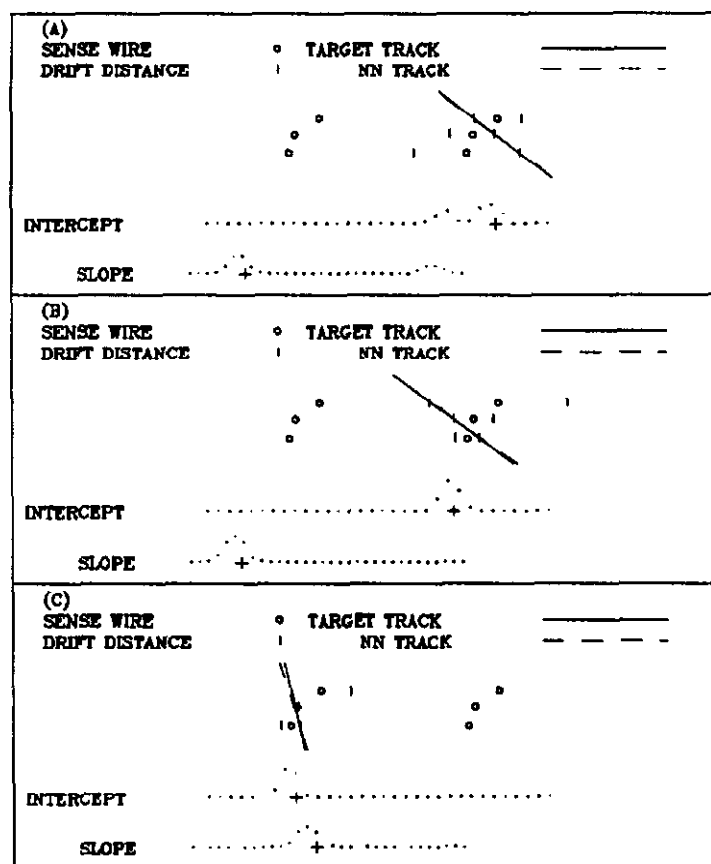


Figure 1: Simulated tracks in a D0 type muon drift chamber. Sense wires are shown; field and ground planes omitted. Tracks produce drift time signals which are shown as vertical bars to sides of wires. The net output units are activated as shown (see fig. 2). The position of the largest activated units gives the net intercept and slope. The + indicates target values.

# NEURAL NETWORK FOR D0 MUON CHAMBER TRACKING

Input = 3 Drift times + 3 signal transit times

Output = 32 0.63cm bins from -0cm to +20cm  
+ 32 0.07rad bins from -1.0rad to 1.2rad

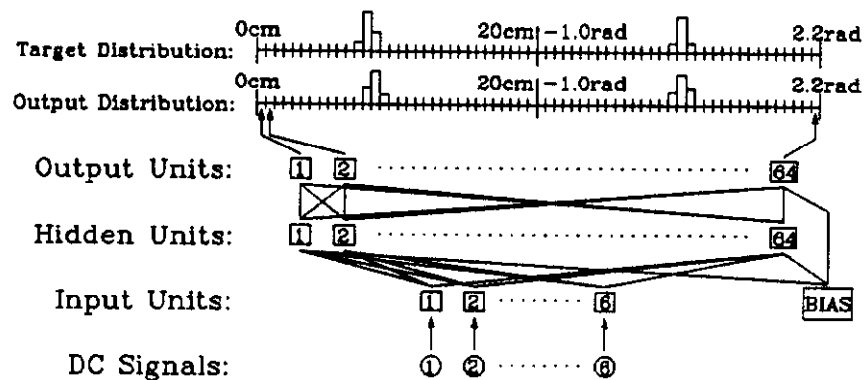


Figure 2: The neural network architecture used to determine the slope and intercept from signals produced by tracks traversing a 6 cell portion of a D0 type muon drift chamber. For clarity only some of the connections are shown.

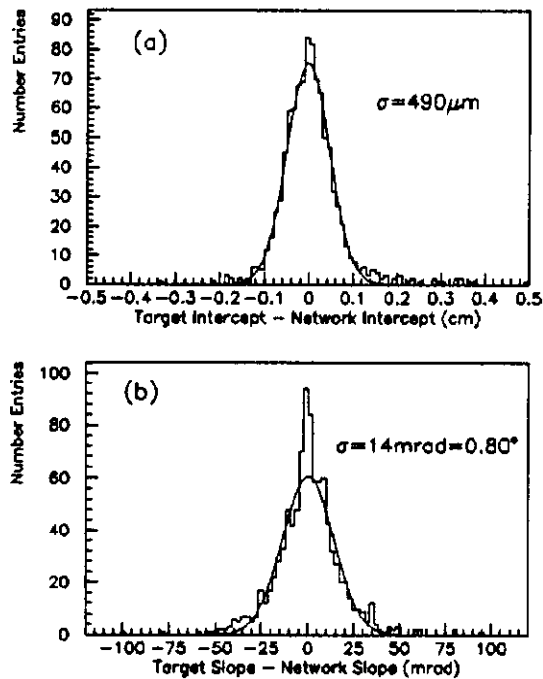


Figure 3: (a) Comparison of target intercept value and that obtained from network output; (b) Comparison of target slope value and that obtained from network output.

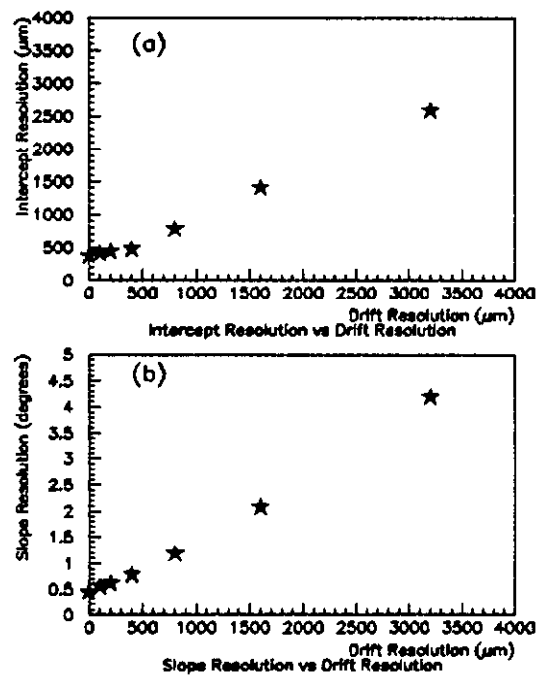


Figure 4: Effects of smearing drift resolution on net tracking resolution. (a) Drift resolution vs net intercept resolution, (b) drift resolution vs net slope resolution.

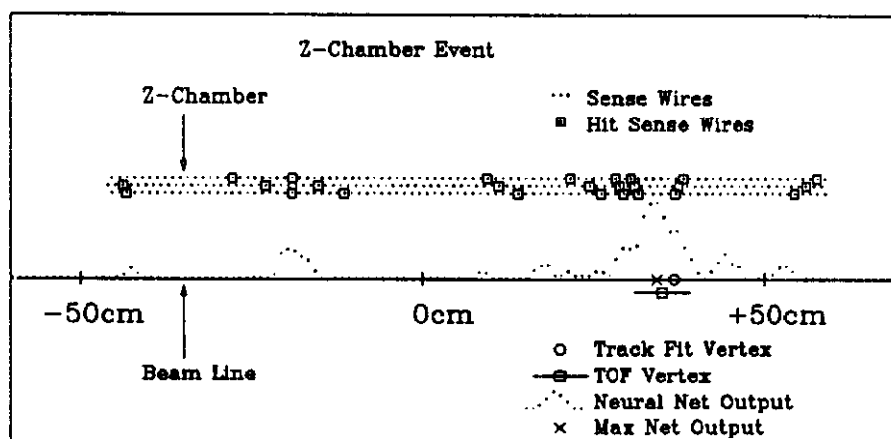


Figure 5: A schematic of the z-chamber with hits from a  $p\bar{p}$  event. Only the sense wires are shown; field wires, beam pipe, etc. are omitted. The event vertices found by three methods are shown: track fitting, time-of-flight, and a neural network simulation.

# NEURAL NETWORK FOR Z-CHAMBER SUB-SECTION VERTEX

Input = 18 Sense Wire Times Normalized to 1.0

Output = 60 1.0cm Bins from -30cm to +30cm

+ 1 Bin for  $Z < -30\text{cm}$  + 1 Bin for  $Z > +30\text{cm}$

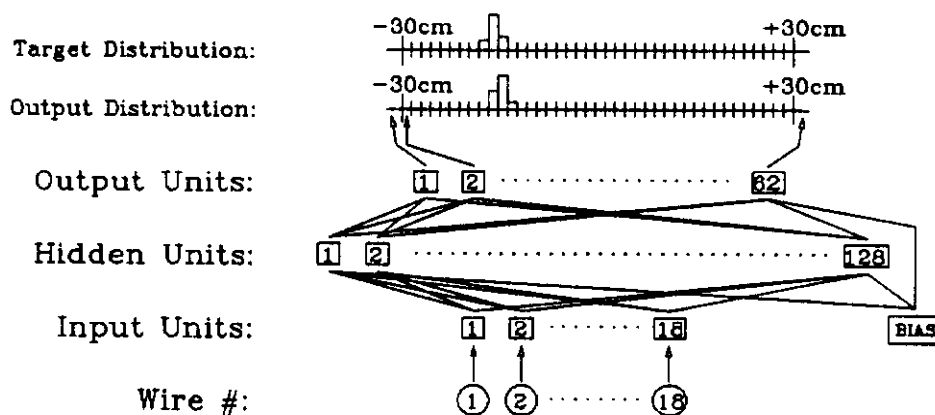


Figure 6: The neural network architecture used to determine the vertex position of tracks in a 18 wire subsection of the z-chamber. Only some of the connections are shown.

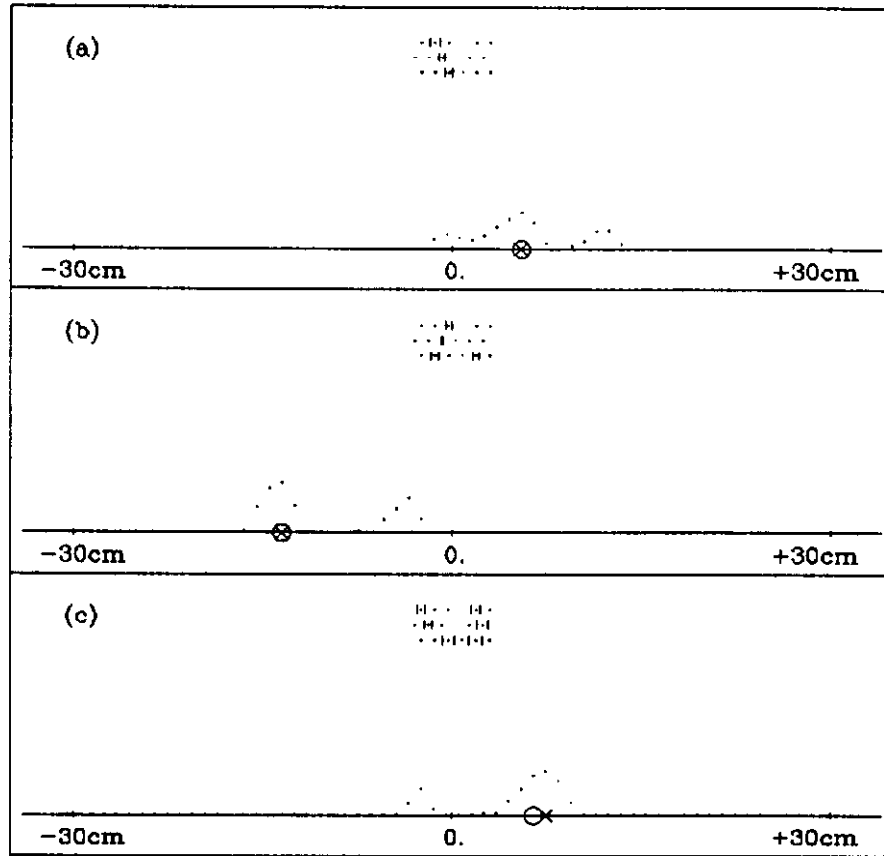


Figure 7: Some examples of tracks in 18 wire subsections of the z-chamber. The vertical bars represent the drift time signals. The relative activations of the net output units are shown. The unit with maximum activation is indicated by the x and its position is compared to that obtained by track fitting.

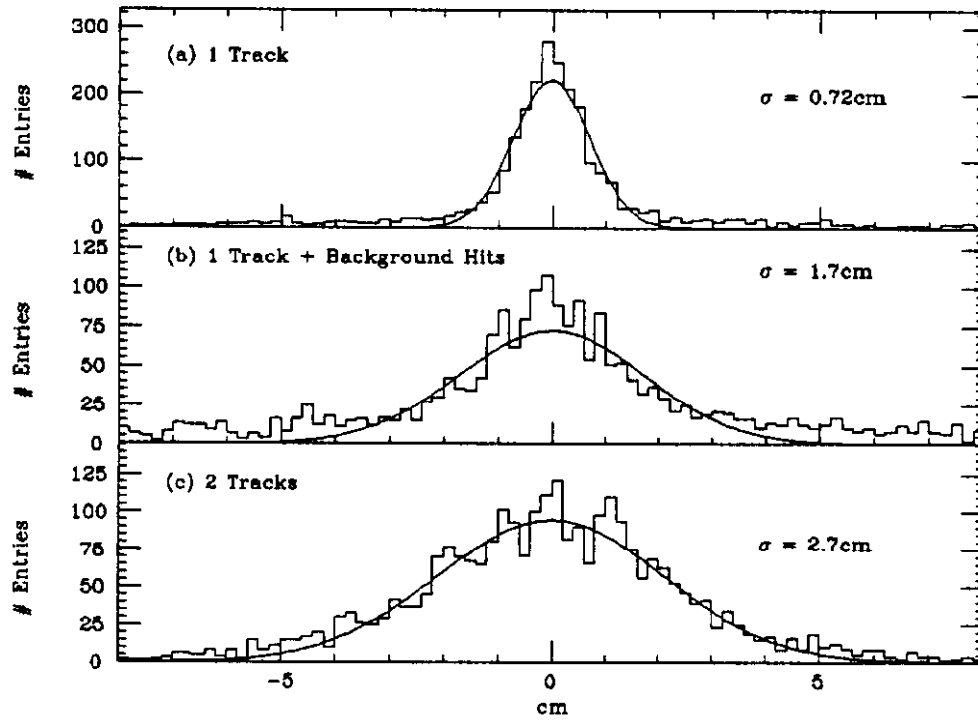


Figure 8: Distributions of the differences between the target track vertices and the net vertices for subsections having hits from (a) single tracks (b) single tracks plus background hits and (c) two tracks.

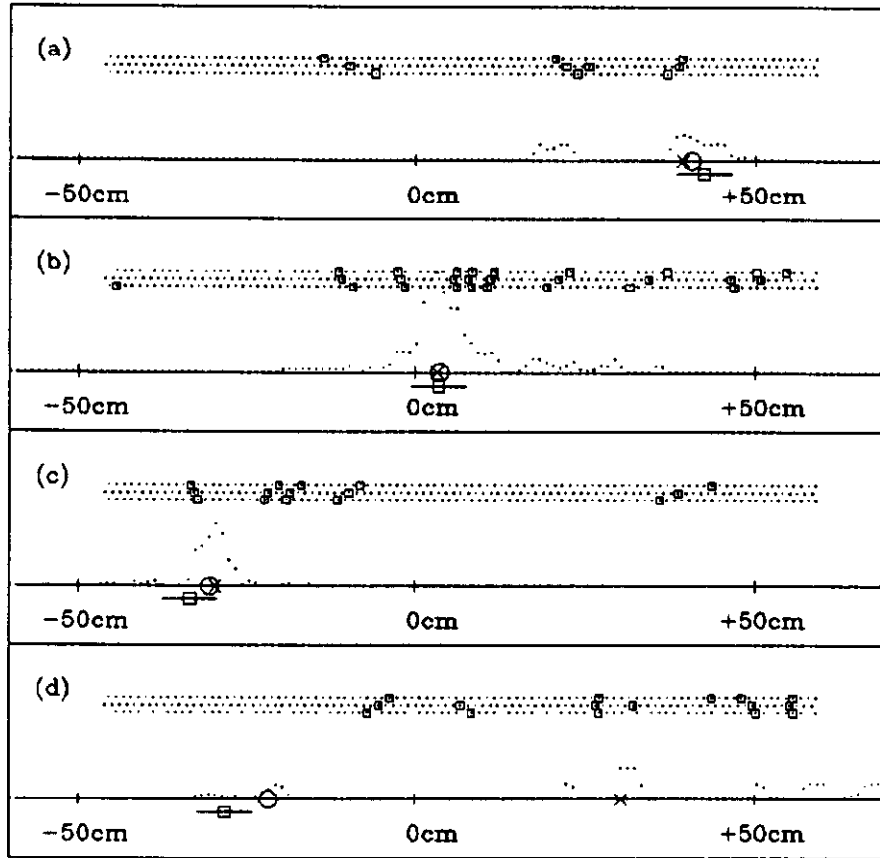


Figure 9: Examples of z-chamber events. The cumulative distribution of adding all subsection neural network outputs is indicated along the beam line. The maximum of the distribution is indicated by an x and compared to the vertex found by track fitting and by the TOF system (symbols same as figure 5).

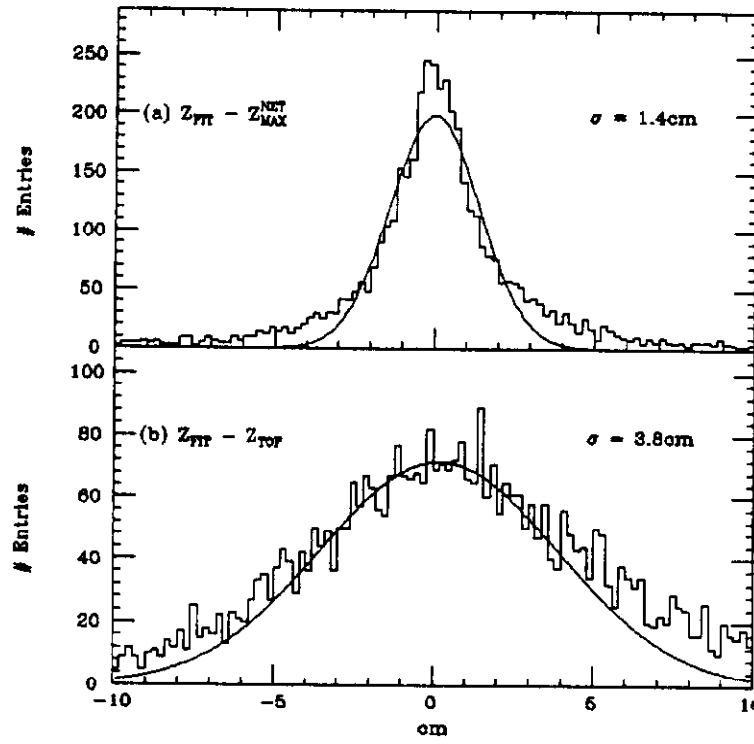


Figure 10: (a) Distribution of differences between the fit event vertex and the neural network event vertex. (b) Differences in fit event vertices and TOF vertices.